PRISM CODE

XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

The full form of XAMPP stands for Cross-platform, Apache, MariaDB(Mysql), PHP and Perl. It is one of the simplest and light-weight local servers that is used to test your website locally. It is an open source platform. This includes X-OS because it works in all major operating systems like Windows, Linux, Mac etc. It includes features like Filezilla, mercury mail, supporting Perl and much more. One of the main advantages is that you can perform as many testing and update the content in your website testing locally. Since it is an open source, you can easily download and install in your system. You can perform a number of testing installing it once.

PHP

PHP is an acronym for "PHP: Hypertext Preprocessor". It is a widely-used, open source scripting language. Its scripts are executed on the server. PHP is free to download and use.

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Basic PHP Syntax

A PHP script can be placed anywhere in the document.

A PHP script starts with **<?php** and ends with **?>.**

PHP statements end with a semicolon (**;**).

PHP Case Sensitivity

PHP is **not** case-sensitive.

However; all variable names are case-sensitive!

PHP Comments

A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code.

Comments can be used to:

- Let others understand your code
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code

PHP supports several ways of commenting:

First way:

```
// This is a single-line comment
```

Second way:

```
# This is also a single-line comment
```

Third way:

```
/* This is a multiple-lines comment block
that spans over multiple
lines */
```

PHP Variables

Variables are "containers" for storing information.

Creating (Declaring) PHP Variables

In PHP, a variable starts with the **$** sign, followed by the name of the variable.
When you assign a text value to a variable, put quotes around the value.
Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

Rules for Declaring a Variable

- A variable starts with the $ sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive ($age and $AGE are two different variables)

Output Variables

The PHP **echo** statement is often used to output data to the screen.

```php
<?php
$txt = "Amazon Web Services";
echo "I love $txt!";
?>
```

I love Amazon Web Services!

PHP is a Loosely Typed Language

In the example above, notice that we did not have to tell PHP which data type the variable is.

PHP automatically associates a data type to the variable, depending on its value. Since the data types are not set in a strict sense, you can do things like adding a string to an integer without causing an error.

PHP Conditional Statements

Very often when you write code, you want to perform different actions for different conditions. You can use conditional statements in your code to do this.

In PHP we have the following conditional statements:

- **if statement -** executes some code if one condition is true
- **if...else statement -** executes some code if a condition is true and another code if that condition is false
- **if...elseif...else statement -** executes different codes for more than two conditions
- **switch statement -** selects one of many blocks of code to be executed

If Statement

The **if** statement executes some code if one condition is true.

```
if (condition) {
    code to be executed if condition is true;
}
```

If-Else Statement

The **if...else** statement executes some code if a condition is true and another code if that condition is false.

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

Switch Statement

The switch statement is used to perform different actions based on different conditions.

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

PHP Loops

Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

- **while -** loops through a block of code as long as the specified condition is true
- **do...while -** loops through a block of code once, and then repeats the loop as long as the specified condition is true
- **for -** loops through a block of code a specified number of times
- **foreach -** loops through a block of code for each element in an array

While Loop

The **while** loop executes a block of code as long as the specified condition is true.

```
while (condition is true) {
    code to be executed;
}
```

Do - While Loop

The **do...while** loop - Loops through a block of code once, and then repeats the loop as long as the specified condition is true.

```
do {
    code to be executed;
} while (condition is true);
```

For Loop

The **for** loop - Loops through a block of code a specified number of times.

```
for (init counter; test counter; increment counter) {
    code to be executed for each iteration;
}
```

Foreach Loop

The **foreach** loop - Loops through a block of code for each element in an array.

```
foreach ($array as $value) {
  code to be executed;
}
```

PHP Arrays

An array stores multiple values in one single variable. An array is a special variable, which can hold more than one value at a time.

```
<?php
$tech = array("HTML", "CSS", "PHP");
echo "I like " . $tech[0] . ", " . $tech[1] . " and " . $tech[2] . ".";
?>
```

I like HTML, CSS and PHP.

Important array functions

array_push( ) : Inserts one or more elements to the end of an array

```
<?php
$a=array("HTML","CSS");
array_push($a,"PHP");
print_r($a);
?>
```

Array ([0] => HTML [1] => CSS [2] => PHP)

array_pop ( ) : Deletes the last element of an array

```
<?php
$a=array("HTML","CSS","PHP");
array_pop($a);
print_r($a);
?>
```

Array ([0] => HTML [1] => CSS)

PHP Global

Variables - Superglobals

Some predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_COOKIE
- $_SESSION

What is JSON?

JSON stands for JavaScript Object Notation, and is a syntax for storing and exchanging data.

Since the JSON format is a text-based format, it can easily be sent to and from a server, and used as a data format by any programming language.

PHP and JSON

PHP has some built-in functions to handle JSON.

- json_encode()

```php
<?php
$age = array("Jeevesh"=>24, "Piyush"=>21, "Pragya"=>21);

echo json_encode($age);
?>
```

```
{"Jeevesh":24,"Piyush":21,"Pragya":21}
```

- json_decode()

```php
<?php
$jsonobj = '{"Jeevesh":24,"Piyush":23,"Pragya":21}';

var_dump(json_decode($jsonobj));
?>
```

```
{ ["Jeevesh"]=> int(24) ["Piyush"]=> int(23) ["Pragya"]=> int(21) }
```